



Les Tables de vérité en braille

Baptiste Mèlès

► To cite this version:

| Baptiste Mèlès. Les Tables de vérité en braille. 2011. hal-01225178

HAL Id: hal-01225178

<https://hal.science/hal-01225178>

Preprint submitted on 5 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Les tables de vérité en braille *

Baptiste Mèlès

11 février 2011

L'intérêt des méthodes de calcul logique en général est de ménager notre mémoire en nous épargnant un calcul mental trop complexe. De ce point de vue, les tables de vérité sont particulièrement inadaptées pour les personnes non-voyantes : elles exigent de mémoriser de longues formules et de se déplacer constamment dans de grands tableaux, chaque déplacement pouvant être source d'erreurs. Aussi allons-nous proposer ici une méthode transposant les tables de vérité en braille, d'une manière qui soit à la fois facilement lisible et aisément calculable, en limitant considérablement le nombre de déplacements.

1 Tables de vérité de quatre lignes

1.1 Codage des distributions de valeurs de vérité


















L'idée fondamentale de ce système consiste à représenter par un seul signe braille une colonne entière de quatre lignes. Une seule ligne de braille codant une table de vérité de quatre lignes, le nombre de déplacements devient presque nul. De tabulaire, le calcul devient linéaire.

Une colonne de quatre lignes contient quatre éléments, qui correspondent à n'importe quelle combinaison possible du vrai (V) et du faux (F) : VVVV, VVVF, VVFF, VFVV, VFVF, VFFV, VFFF, FVVV, FVVF, FVFF, FFVV, FFVF, FFFV, FFFF. Pour les représenter, nous allons nous représenter un signe braille comme composé de quatre barres verticales successives, à savoir :

- d'abord les points 1 et 2, qui composent le signe « b » $\begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \end{smallmatrix}$
- ensuite les points 2 et 3, qui composent le signe « ; » $\begin{smallmatrix} \bullet & \bullet \\ \vdots & \bullet \end{smallmatrix}$
- puis les points 4 et 5, qui composent le signe $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$
- enfin les points 5 et 6, qui composent le signe $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$

*L'auteur tient à remercier Laurence Gaston, dont l'expérience du braille a permis de créer un système qui soit le plus lisible possible, et Michel Serfati, qui l'a sensibilisé à l'enjeu intuitif des notations symboliques.

En lisant ces « traits » dans l'ordre où l'on lit les points du braille, on parcourt d'un seul coup une colonne de quatre valeurs de vérité :

- VVVV est codé par « é » 
- VVVF est codé par « q » 
- VVFF est codé par « à » 
- VVFF est codé par « l » 
- VFVV est codé par « ï » 
- VFVF est codé par « g » 
- VFFV est codé par « ü » 
- VFFF est codé par « b » 
- FVVV est codé par « ù » 
- FVVF est codé par « t » 
- FVFF est codé par « ” » 
- FVFF est codé par « ; » 
- FFVV est codé par l’italique 
- FFVF est codé par le signe 
- FFFV est codé par le signe 
- FFFF devrait être codé par l’espace  mais il sera souvent plus pratique de le représenter par exemple par « e »  afin de réserver les espaces pour aérer le texte.

On peut ainsi représenter :

- la proposition « p » (VFFF) par $\begin{smallmatrix} \bullet \\ \bullet \\ \bullet \\ \bullet \end{smallmatrix}$
- la proposition « q » (VFVF) par $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$
- la proposition « p et q » (VFFF) sera alors représentée par $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$
- la proposition « p ou q » (VVVF) par $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$
- « p implique q » (VFVV) par $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$
- etc.

1.2 Codage des connecteurs

On peut noter les connecteurs logiques de la manière suivante :

- la conjonction par un point \bullet .

- la disjonction par « v » $\begin{smallmatrix} \bullet \\ \vdots \\ \bullet \end{smallmatrix}$
- l'implication par « o » $\begin{smallmatrix} \bullet \\ \vdots \\ \bullet \end{smallmatrix}$
- l'équivalence par « : » $\begin{smallmatrix} \bullet \\ \vdots \\ \bullet \end{smallmatrix}$
- la négation par « n » $\begin{smallmatrix} \bullet \\ \vdots \\ \bullet \end{smallmatrix}$
- les parenthèses par les parenthèses braille $\begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix}$ et $\begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix}$

Ainsi, les parenthèses sont notées en braille par des parenthèses ; la conjonction est notée, comme chez certains auteurs, par un point ; la disjonction, comme toujours, par un « v » ; l'implication par un signe asymétrique, qui ressemble un peu à une flèche ; et l'équivalence par un signe symétrique.

L'autre intérêt de cette notation est qu'à quelques nuances près (notamment le « o » pour l'implication), une fois imprimée en noir, une formule écrite dans ce langage est lisible pour un logicien ordinaire. Par exemple, la formule « non (p et q) si et seulement si non p ou non q » sera imprimée « n (p.q) : np v nq ».

Enfin, on peut choisir d'inscrire toutes les formules ou démonstrations entre des guillemets américains $\begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix}$. Par chance, cette notation n'interférera pas avec la proposition « non q ».

Voici quelques exemples de cette notation.

- (p implique q) ou (q implique p) :
 $\begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet \\ \vdots \\ \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet \\ \vdots \\ \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet \\ \vdots \\ \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix}$
- non (p et q) si et seulement si non p ou non q :
 $\begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix}$

1.3 Règles des connecteurs logiques

Maintenant que nous avons codé en braille la formule à étudier, nous pouvons calculer sa table de vérité en braille. Il suffit pour cela de connaître quelques règles de calcul, une par connecteur logique.

1.3.1 Conjonction

La conjonction correspond à l'intersection des deux signes braille. Par exemple, la proposition « p et q », notée $\begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix}$, ne conserve que les points qui figurent à la fois dans les deux signes braille, c'est-à-dire dans le premier *et* dans le deuxième :

$\begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix} \begin{smallmatrix} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{smallmatrix}$

1.3.2 Disjonction

La disjonction correspond à l'union des deux signes : on conserve dans le signe final tous les points qui figurent dans le premier signe *ou* dans le deuxième. Voyons l'exemple de « p ou q » :

$\begin{array}{cccc} \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} \\ \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & & \end{array}$

1.3.3 Implication

L'implication est, comme dans les tables de vérité traditionnelles, le connecteur le plus complexe. Chacun des quatre traits du signe braille final doit être validé si son homologue dans le premier signe est inférieur ou égal à son homologue du deuxième signe. Soit par exemple la proposition « q implique q » :

$\begin{array}{ccccc} \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} \end{array}$

Le premier « trait » de $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$, c'est-à-dire les points 1 et 2, est plein, de même que son homologue dans $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$: il lui est inférieur ou égal, donc le résultat doit contenir un trait plein à cet endroit. Il n'en va pas de même du deuxième trait, composé des points 2 et 3 : ce trait est plein dans $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$ mais vide dans $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix}$: le premier n'étant pas inférieur ou égal au second, le résultat doit être vide à cet endroit. En continuant de la sorte pour les troisième et quatrième traits, on parvient au résultat suivant :

$\begin{array}{cccc} \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} \\ \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & & \end{array}$

Une manière de simplifier les calculs peut être de passer par une traduction de l'implication en disjonction : « p implique q » est en effet équivalent à « non p ou q ». On pourra donc procéder au calcul suivant :

$\begin{array}{cccc} \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} \\ \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} \\ \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} \\ \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & \begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} & & \end{array}$

1.3.4 Négation

Pour calculer la négation, il suffit d'inverser la valeur de chaque point, en remplaçant les vides par des pleins et les pleins par des vides, puis en rajoutant en cas de besoin les points 2 et 5 pour obtenir de véritables traits. Voici l'exemple de « non p » :

⠠⠏ ⠠⠑ ⠠⠒
⠠⠑ ⠠⠒

Et l'exemple de « non q » :

⠠⠏ ⠠⠑ ⠠⠒
⠠⠑ ⠠⠒

1.3.5 Équivalence

Pour calculer l'équivalence, il suffit de valider tous les points identiques dans le premier et le second signe : un trait est plein dans le résultat s'il est plein dans les deux signes, ou vide dans les deux signes. Prenons l'exemple de « p si et seulement si q », où le premier trait est plein dans les deux cas, le dernier vide dans les deux cas :

⠠⠏ ⠠⠑ ⠠⠒ ⠠⠑
⠠⠑ ⠠⠒

1.4 Méthode de calcul









La méthode de calcul des tables de vérité en braille est la même que pour les tables de vérité traditionnelles : on calcule d'abord les plus petites sous-formules, puis on remonte progressivement vers la formule globale. On réécrit ligne après ligne toute la formule, en remplaçant les sous-formules par leur table de vérité.

Voici par exemple la table de vérité de l'une des lois de De Morgan :

⠠⠏ ⠠⠑ ⠠⠒ ⠠⠓ ⠠⠔ ⠠⠕ ⠠⠖ ⠠⠗ ⠠⠘ ⠠⠙ ⠠⠚ ⠠⠛
⠠⠑ ⠠⠒ ⠠⠓ ⠠⠔ ⠠⠕ ⠠⠖ ⠠⠗
⠠⠑ ⠠⠒ ⠠⠓ ⠠⠔ ⠠⠕ ⠠⠖
⠠⠑ ⠠⠒ ⠠⠓ ⠠⠔
⠠⠑ ⠠⠒ ⠠⠓
⠠⠑ ⠠⠒

La formule est donc une tautologie.

A 10x10 grid of dots representing a sparse matrix. The dots are arranged in a pattern that is mostly diagonal, with some off-diagonal elements. The dots are black and the background is white.

p	q	p et q	non (p et q)	non p	non q	non p ou non q	A
1	1	1	0	0	0	0	1
1	0	0	1	0	1	1	1
0	1	0	1	1	0	1	1
0	0	0	1	1	1	1	1
							

Cette méthode de calcul de tables de vérité en braille est, telle quelle, soumise à une limitation importante, car elle ne permet de calculer que des tables de quatre lignes, donc de ne résoudre que les formules contenant une ou deux variables propositionnelles, typiquement nommées « p » et « q ». Que faire quand une formule contient trois variables propositionnelles ou davantage ? On peut envisager deux méthodes de résolution, naturellement

équivalentes.

La première méthode serait de réaliser deux tables de vérité de quatre lignes, une dans le cas où « r » soit vraie, l'autre dans le cas où cette proposition soit fausse. Mais on oublierait facilement de réaliser la deuxième table ; aussi cette méthode nous paraît-elle dangereuse.

La seconde méthode, que nous préconisons, consiste à dédoubler chaque signe braille pour représenter les huit combinaisons possibles avec trois variables propositionnelles, à le quadrupler pour représenter les seize combinaisons possibles avec quatre variables propositionnelles, etc. Supposons que l'on veuille calculer la table de vérité de la formule suivante :

p et $(q$ ou $r)$ si et seulement si $(p$ et $q)$ ou $(p$ et $r)$

Alors on représentera :

- la proposition « r » par le couple de signes $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} \quad \begin{smallmatrix} \bullet \\ \bullet \end{smallmatrix}$, dont le deuxième signe permet de réserver les espaces pour aérer le texte : ici aussi, le « e » représente la contradiction, c'est-à-dire la séquence FFFF ;
- la proposition « p » par le couple $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} \quad \begin{smallmatrix} \bullet \\ \bullet \end{smallmatrix}$;
- la proposition « q » par le couple $\begin{smallmatrix} \bullet & \bullet \\ \bullet & \bullet \end{smallmatrix} \quad \begin{smallmatrix} \bullet \\ \bullet \end{smallmatrix}$;

On calculera donc la table de vérité suivante :

A 10x10 grid of dots representing a sparse matrix. The dots are arranged in a pattern that is mostly diagonal, with some off-diagonal elements, suggesting a sparse matrix structure.

La formule est donc une tautologie.

La table de vérité de cette formule A est la suivante :

p	q	r	q ou r	p et (q ou r)	...
1	1	1	1	1	...
1	1	0	1	1	...
1	0	1	1	1	...
1	0	0	0	0	...
0	1	1	1	0	...
0	1	0	1	0	...
0	0	1	1	0	...
0	0	0	0	0	...
⠠p	⠠q	⠠r	⠠q ou r	⠠p et (q ou r)	...
...	p et q	p et r	(p et q) ou (p et r)	A	
...	1	1	1	1	
...	1	0	1	1	
...	0	1	1	1	
...	0	0	0	1	
...	0	0	0	1	
...	0	0	0	1	
...	0	0	0	1	
...	0	0	0	1	
...	⠠p et q	⠠p et r	⠠(p et q) ou (p et r)	⠠A	

De la même manière, on pourra coder des tables de vérité d'une longueur quelconque. Par exemple, si la formule contient quatre variables propositionnelles, on pourra les représenter de la manière suivante :

- la proposition « p » sera notée ⠠p
- la proposition « q » sera notée ⠠q
- la proposition « r » sera notée ⠠r
- la proposition « s » sera notée ⠠s

Certes, la formule devient bientôt difficilement manipulable ; mais une table de vérité composée de seize lignes n'est pas moins fastidieuse.

3 Remarque

On aurait pu penser représenter chaque lettre propositionnelle par un point des six points du braille, et ainsi coder en un seul signe braille toute combinaison de six variables propositionnelles, c'est-à-dire n'importe quelle colonne d'une table de vérité de 64 lignes, selon le codage suivant :

- la proposition « p » serait notée ⠠
- la proposition « q » serait notée ⠠

- la proposition « r » serait notée $\begin{smallmatrix} \cdot \cdot \\ \cdot \cdot \end{smallmatrix}$
- la proposition « s » serait notée $\begin{smallmatrix} \cdot \cdot \\ \cdot \cdot \end{smallmatrix}$
- la proposition « t » serait notée $\begin{smallmatrix} \cdot \cdot \\ \cdot \cdot \end{smallmatrix}$
- la proposition « u » serait notée $\begin{smallmatrix} \cdot \cdot \\ \cdot \cdot \end{smallmatrix}$

Mais ce système aurait le grand inconvénient d'être presque illisible pour un lecteur de braille, pour qui il est parfois difficile de connaître la position précise d'un point isolé ; et le gain serait très faible, étant donné que les tables de vérité étudiées en cours de logique dépassent en pratique très rarement les quatre ou huit lignes.